

Henri Nikka

# Suunnitteluprosessin määrittäminen ohjelmisto- alan yritykselle

Metropolia Ammattikorkeakoulu

Medianomi (AMK)

Viestinnän koulutusohjelma

Opinnäytetyö

14.5.2017

Tekijä(t) Otsikko Sivumäärä Aika	Henri Nikka Suunnitteluprosessin määrittäminen ohjelmisto-alan yritykselle 26 sivua + 1 liite 14.5.2017
Tutkinto	Medianomi (AMK)
Koulutusohjelma	Viestinnän koulutusohjelma
Suuntautumisvaihtoehto	Digitaalinen viestintä
Ohjaaja(t)	Lehtori Juhana Kokkonen
<p>Tämän opinnäytetyön tarkoituksena on määrittää suunnitteluprosessi ohjelmistoalan yritykselle. Työn toimeksiantaja on ohjelmistoalan yritys Kisko Labs Oy, joka tuottaa verkkopalveluita ja mobiilisovelluksia eri toimialoille. Suunnitteluprosessin määrittämisen tavoitteena on parantaa suunnittelutyön kommunikointia Kisko Labsin asiakkaille sekä sen työntekijöille sisäisesti.</p> <p>Työssä määritellään suunnitteluprosessi perustuen yrityksen olemassa oleviin työtapoihin. Prosessi muotoillaan myös sopimaan osaksi yrityksen ohjelmistokehitystä, joka noudattaa ketterän ohjelmistokehityksen periaatteita. Ketterän ohjelmistokehityksen menetelmiä tarkastellaan lähemmin, jotta suunnitteluprosessi ei ole ristiriidassa niiden kanssa.</p> <p>Prosessin määrittämiseksi kartoitettiin yrityksen työskentelytapoja tekemällä sisäinen haastattelututkimus työntekijöille. Haastattelun tulokset analysoitiin ja löydösten pohjalta järjestettiin kaksi työpajaa lopullisen prosessin määrittämiseksi.</p> <p>Työn lopuksi esitetään syntynyt prosessikuvaus. Työn tuloksena ei syntynyt täysin uutta prosessia, vaan prosessikuvaus perustuen yrityksen olemassa oleville työskentelytapoille. Prosessin määrittäminen auttoi tarkkailemaan yrityksen suunnittelutyötä tarkemmin, ja se auttoi löytämään parannusehdotuksia.</p>	
Avainsanat	suunnittelu, prosessi, ketterät menetelmät, ohjelmistokehitys

Author(s) Title	Henri Nikka Defining Design Process for Software Development Company
Number of Pages Date	26 pages + 1 appendix 14 May 2017
Degree	Bachelor of Arts and Culture
Degree Programme	Degree Programme in Media
Specialisation option	Digital Media
Instructor(s)	Juhana Kokkonen, Senior Lecturer
<p>The purpose of this study is to define a design process for a Finnish software development company, Kisko Labs Oy. Kisko Labs builds web services and mobile applications for different lines of business. The goal of the design process is to improve the communication, between designers and clients, and internally among team members.</p> <p>The design process defined in this study was built around the company's well established work habits, and it was designed to integrate seamlessly with the company's agile software development process. Agile software development methods are observed closely to achieve this.</p> <p>The main research method is an interview survey. The goal of the interviews was to understand the existing ways of working. The results were analysed and based on knowledge gathered from the interviews, two workshops were organised. The purpose of the workshops was to help finalize the design process.</p> <p>The results are presented in the shape of a well-defined design process. The outcome is not a completely new process, but rather a process that depicts and explains the company's design methods. The process helps communicate and improve design work further.</p>	
Keywords	design, process, agile methods, software development

## Sisällys

1	Johdanto	1
2	Suunnittelutyö Kisko Labs Oy:ssä	2
2.1	Yleisesti Kisko Labsistä	2
2.2	Tarpeet	2
2.3	Tavoitteet	3
3	Ohjelmistokehitysmenetelmät	4
3.1	Ketterä ohjelmistokehitys	4
3.2	Scrum	5
3.3	Extreme programming	6
3.4	Ohjelmistokehitys Kisko Labsillä	7
4	Prosessin määrittäminen	8
4.1	AloitUS	8
4.2	Haastattelututkimus	10
4.3	Haastattelun tulokset	11
4.4	Haastattelun tulosten analysointi	12
4.5	Ensimmäinen työpaja	13
4.6	Toinen työpaja	15
5	Johtopäätökset	19
5.1	Suunnitteluprosessin kuvaus	19
5.2	Integroituminen kehitysprosessiin	20
5.3	Jatkotoimenpiteet	21
6	Yhteenveto	23
	Lähteet	25
	Liitteet	
	Liite 1. Puolistrukturoidun haastattelun kysymykset	

## 1 Johdanto

Ohjelmistokehitykseen voi työn luonteesta riippuen liittyä monenlaista suunnittelutyötä. Ohjelmointityössä suunnitellaan tietokantoja ja määritetään teknistä arkkitehtuuria. Tällainen suunnittelu helpottaa ohjelmiston koodausta, mutta koodi ei suoranaisesti näy loppukäyttäjälle. Jos käyttäjät ovat vuorovaikutuksessa ohjelmiston kanssa, tarvitaan muitakin suunnittelukeinoja. Tämän työn rajauksessa keskitytään verkkopalveluiden suunnittelussa käytettyihin keinoihin, jotka näkyvästi vaikuttavat loppukäyttäjän kokemukseen. Tärkeimmät suunnittelukeinot ovat käyttökokemussuunnittelu ja käyttöliittymäsuunnittelu, joissa tärkeimmässä roolissa ovat käyttäjät.

Tämän opinnäytetyön tavoitteena on määrittää suunnitteluprosessi ohjelmistoalan yritykselle. Tutkimuksen aihe on valittu työn tilanteen yrityksen tarpeiden perusteella. Yrityksellä ei ollut lähtötilanteessa ollenkaan suunnitteluprosessia ja ongelmana oli prosessin vaiheiden kommunikointi asiakkaille ja sisäisesti. Tarkoitus ei kuitenkaan ollut kehittää täysin uutta prosessia, vaan kartoittaa nykyiset toimintatavat ja määritellä havaintojen perusteella konkreettinen prosessi. Työn tilaaja on Kisko Labs Oy, vuonna 2007 perustettu pieni ohjelmistoalan yritys. Tämän opinnäytetyön kirjoittaja on itse töissä suunnittelijana Kiskolla (Kisko Labs Oy lyhennetty myös myöhemmin muotoon Kisko) ja omat havainnot vaikuttivat myös lopputulokseen.

Tässä opinnäytetyössä käydään aluksi läpi Kiskon yleisiä toimintaperiaatteita ja tarpeita. Kolmannessa luvussa käydään läpi yrityksen ohjelmistokehityksessä käytettäviä ketteriä menetelmiä, sillä ne ovat täysin vuorovaikutuksessa suunnittelutyön kanssa. Neljännessä luvussa aloitetaan suunnitteluprosessin määrittäminen. Prosessin määrittäminen etenee kronologisesti sen mukaan, missä vaiheet oikeasti toteutettiin. Prosessin selvittämiseksi käytetään haastattelututkimusta ja sen tulosten pohjalta järjestettyjä työpajoja. Lopuksi käydään läpi syntynyt prosessi ja pohditaan sen onnistumista ja jatkokehitystarpeita.

## 2 Suunnittelutyö Kisko Labs Oy:ssä

### 2.1 Yleisesti Kisko Labsistä

Kisko Labs Oy on vuonna 2007 perustettu pieni ohjelmistoyritys. Se toteuttaa verkkopalveluita ja mobiilisovelluksia käyttäen pääsääntöisesti Ruby on Rails -teknologiaa. Kiskon asiakkaat vaihtelevat pienistä startup-yrityksistä suuriin organisaatioihin. Myös asiakkaiden toimialat ovat varsin erilaisia mm. hyvinvointi, taloudenhoito ja liiketoiminnan johtaminen. Kiskon tarkoituksena ei ole vain tarjota kehitystyötä, vaan auttaa asiakasta löytämään ratkaisuja liiketoiminnan kehittämiseksi digitaalisten palveluiden avulla.

Työ on projektiluontoista, ja projektit kestävät parista viikosta useampiin kuukausiin. Pienemmissä projekteissa tiimin koko on yhdestä kolmeen henkilöä ja tiimissä on monesti vain yksi suunnittelija. Suunnittelijan rooli on moninainen, koska hän voi vastata yksin isommankin projektin suunnittelutyöstä. Työvaiheisiin voi kuulua mm. konseptisuunnittelua, käyttäjätutkimusta, informaatioarkkitehtuuria, rautalankojen piirtämistä ja graafista suunnittelua. Suunnittelijan toimenkuvaan kuuluu myös käyttöliittymän lopullinen toteutus, ja kaikki suunnittelijat tekevät front-end-kehitystä (selaimessa pyörivän ohjelmistokoodin toteutus HTML:n, CSS:n ja Javascriptin avulla).

### 2.2 Tarpeet

Projektityöt ovat pääsääntöisesti ohjelmistokehitystä, joissa lopputuotteena on verkkopalvelu, mobiilisovellus tai ylipäättänsä mikä tahansa ohjelmistotuote. Ohjelmistokehittäjät tuottavat pääsääntöisesti koodia, jonka toiminnallisuus ja arvo on helppo todentaa testaamalla ohjelmistoa. Suunnittelijat tuottavat myös koodia, mutta työssä on paljon muitakin vaiheita kuin käyttöliittymän koodaus. Näiden muiden vaiheiden tuotoksia on vaikeampi arvioida, sillä niistä ei välttämättä synny mitään konkreettista lopputuotteen. Ne ovat kuitenkin monesti välttämättömiä vaiheita, joiden avulla päästään lähemmäksi lopputulosta. Esimerkkinä tällaisesta vaiheesta on käyttötapausten kartoittaminen, jonka tuotoksena syntyy dokumentaatiota. Dokumentaatio voi tuoda arvoa asiakkaalle, mutta asiakas ei ole välttämättä tilannut dokumentaatiota vaan toimivan ohjelmiston. Pyrkimys on tuottaa vain tarpeellista dokumentaatiota, ja asiakkaan on ymmärrettävä siihen käytetyt resurssit ja dokumentaation tuoma arvo (Sharma 2013).

Kiskon kehitystyö noudattaa ketterän ohjelmistokehityksen menetelmiä, ja kehityksen vaiheet ovat yleensä selkeitä. Yksinkertaistettuna voidaan sanoa, että kehityksessä tuotetaan aina koodia. Suunnittelu on hyvin integroitu ketterään ohjelmistokehitykseen, ja ne ovat riippuvaisia toisistaan. Ongelmana on kuitenkin se, että työn tuottama arvo ei erotu edukseen kehitystyön rinnalla. Erillisen suunnitteluprosessin määrittämisen tarkoituksena on korostaa suunnittelutyön tuomaa arvoa. Päämääränä ei kuitenkaan ole erottaa sitä täysin erilliseksi prosessiksi vaan määritellä, miten se etenee kehitystyön rinnalla.

Suunnittelun työvaiheet vaihtelevat paljon projektin tarpeista riippuen. Projekti voi lähteä pelkästä ideasta, jolloin konseptoidaan palvelu alusta loppuun asti. Toisinaan projektina voi olla yksinkertaisen prototyypin toteutus, jolloin vaiheita on vain muutama. Tämän takia suunnitteluprosessin pitää pystyä muuttumaan aina tarpeiden mukaan. Prosessia ei kannatakaan mallintaa yksityiskohtaisella tasolla, jos se sisältää paljon epävarmoja muuttujia (Martinsuo & Blomqvist 2010, 3-4).

### 2.3 Tavoitteet

Prosessiajattelun tavoitteena on yrityksen tai organisaation tuloksellisuuden kehittäminen. Tuloksellisuus voi tarkoittaa eri yrityksille eri asioita; tärkeätä on selvittää päämäärät, joita kohden pyrkii. (Martinsuo ym. 2010, 3.) Tässä tutkimuksessa tärkeimpänä tavoitteena on parantaa suunnittelutyön vaiheiden kommunikointia sekä asiakkaalle että yrityksen työntekijöille sisäisesti.

Kommunikoinnin odotetaan helpottavan asiakkaan kanssa yhdessä työskentelyä, lisäävän luottamusta ja ohjaavan palautteen antamisessa. Tarjousvaiheessa olisi helpompi perustella asiakkaalle työmääräarviota, kun tiedetään tarkemmin, mihin arvio perustuu. Projektin päätyttyä asiakas ymmärtäisi paremmin, mitä arvoa hän on saanut tuotetun koodin lisäksi. Sisäisesti prosessin vaiheiden kommunikointi helpottaisi työntekijöiden siirtoa keskeneräisiin projekteihin ja antaisi paremman käsityksen projektin etenemisestä. Prosessi helpottaisi myös työntekijöiden työn vaiheistusta ja tekisi aikatauluarvioista tarkempia. Tavoitteiden toteutuessa voitaisiin pienentää riskejä ja säästää aikaa nopeuttamalla projektien etenemistä.

### 3 Ohjelmistokehitysmenetelmät

#### 3.1 Ketterä ohjelmistokehitys

Perinteinen tapa hallita ohjelmistotuotantoa on vesiputousmalli, jossa kehitys alkaa vaatimusten määrittelystä ja loppuu ylläpitovaiheeseen. Vesiputousmallissa keskitytään ohjelmiston määrittelyyn, dokumentointiin, testaamiseen ja ylläpitoon. Itse kehitystyö tai ohjelmointi jää pienempään rooliin. (Duka 2013, 426.)

Vesiputousmallissa pyritään määrittelemään kaikki mahdolliset ohjelmiston ominaisuudet ja toiminnallisuudet. Vaatimukset dokumentoidaan heti projektin ensimmäisessä vaiheessa, ennen varsinaisen ohjelmointityön alkua. Päämääränä on löytää kaikki tarpeet kerralla, eikä uusia vaatimuksia saa lisätä jälkikäteen. Muutoksia projektin vaatimuksiin tulee kuitenkin väistämättä eteen, ja niiden ennustamisesta tulee hankalaa. Vesiputousmallia yritettiin kehittää eteenpäin, jakamalla projekti pienempiin kokonaisuuksiin. Sama ongelma kuitenkin säilyi, ja muutoksia pyrittiin ennustamaan etukäteen. Ketterät menetelmät kehittyivät ratkaisemaan tämän ongelman. (Cohen & Lindvall & Costa, 2003 3.) ”Ketterät menetelmät kehittyivät hyväksymään muutokset, eivätkä torjumaan niitä” (Williams & Cockburn 2003, 39).

Ketterän ohjelmistokehityksen menetelmät saivat alkunsa vuonna 2001 julkaistussa ketterän ohjelmistokehityksen julistuksessa:

Löydämme parempia tapoja tehdä ohjelmistokehitystä, kun teemme sitä itse ja autamme muita siinä. Kokemuksemme perusteella arvostamme:

- Yksilöitä ja kanssakäymistä enemmän kuin menetelmiä ja työkaluja
- Toimivaa ohjelmistoa enemmän kuin kattavaa dokumentaatiota
- Asiakasyhteistyötä enemmän kuin sopimusneuvotteluja
- Vastaamista muutokseen enemmän kuin pitäytymistä suunnitelmassa

Jälkimmäisilläkin asioilla on arvoa, mutta arvostamme ensiksi mainittuja enemmän. (Beck, Beedle, van Bennekum, Cockburn, Cunningham, Fowler, Grenning, Highsmith, Hunt, Jeffries, Kern, Marick, Martin, Mellor, Schwaber, Sutherland & Thomas 2001.)

Ketterien menetelmien tärkeimmät piirteet ovat yksinkertaisuus ja nopeus. Aluksi toimitetaan vain tärkeimmät toiminnallisuudet, kerätään palautetta ja reagoidaan saatuun pa-



lautteeseen. Yhteenvedona ketterä kehitys on inkrementaalista, suoraviivaista, joustavaa, ja siinä työskennellään yhteistyössä kaikkien osapuolten kanssa. (Abrahamsson, Salo, Ronkainen & Warsta 2002, 17.)

Ketterät menetelmät eivät tarkoita yksittäistä työtapaa, vaan tapoja toteuttaa ketterää ohjelmistokehitystä on useita. Yhteistä näille kaikille tavoille on se, että ne kaikki noudattavat ketterän ohjelmistokehityksen julistusta (cPrime Inc, 2017). Seuraavissa alaluvuissa esitellään pari yleistä menetelmää.

### 3.2 Scrum

Scrum on yksi suosituimmista tavoista toteuttaa ketterää kehitystä. Se on viitekehys, joka noudattaa ketterien menetelmien periaatteita. Scrumissa kehitystyö tehdään lyhyissä iteraatioissa, joita kutsutaan sprinteiksi. Sprinttien tarkoituksena on toimittaa säännöllisin väliajoin pieniä mutta toimivia kokonaisuuksia kehitettävästä ohjelmistosta. Näistä pienistä osista on helpompi kerätä palautetta, ja niiden sisältöä on helpompi arvioida. Sprintit auttavat myös motivoimaan kehittäjiä, sillä onnistumisia tapahtuu useammin, kuin vain projektin lopuksi. (Radigan 2017.)

Sprintti on lyhytkestoinen, enintään kuukauden pituinen jakso. Jakson aikana tuotetaan käyttökelpoinen versio ohjelmistosta, jota voidaan testata. Sprintissä on neljä aktiviteettiä: suunnittelu, päiväpalaveri, katselmointi ja retrospektiivi. Suunnittelussa päätetään, mitä kehitystyötä tehdään jakson aikana. Työ määräytyy sen mukaan, mitä on mahdollista toteuttaa lyhyen jakson sisällä. Päiväpalaveri on 15 minuutin mittainen kokous, jossa päätetään alkavan päivän töistä. Palaverissa käydään läpi edellisen päivän tuotokset ja arvioidaan niiden perusteella seuraavat toimenpiteet. Katselmointi tehdään sprinttijakson loputtua. Siinä käydään läpi tuotettu versio ohjelmistosta, kerätään palautetta ja suunnitellaan koko projektin etenemistä. Katselmointi antaa jo perustaa seuraavan sprintin suunnittelulle. Retrospektiivi pidetään heti katselmoinnin jälkeen, ja siinä tarkastellaan kehitystiimin toimintaa. Ideana on parantaa kehitysprosessia ja ratkaista mahdolliset ongelmat työskentelyn tehokkuudessa ja mielekkyydessä. Koko tiimi osallistuu kaikkiin kokouksiin alusta alkaen. (Schwaber & Sutherland 2016, 7–12.)

Scrum koostuu scrum-tiimeistä, joissa on kolme määrättyä roolia: tuoteomistaja, scrum-master ja kehitystiimi. Tiimit ovat itseohjautuvia ja päättävät itse, miten työ tehdään. Ul-

kopuoliset henkilöt eivät saa vaikuttaa tiimin työskentelyyn. Tuoteomistaja vastaa kehitettävästä tuotteesta. Hän päättää, mitä kehitetään ja missä järjestyksessä, ja muiden on kunnioitettava tätä päätöstä. Tuoteomistaja luo kehitysjonon, joka kertoo mitä ja missä järjestyksessä kehitetään. Hän voi halutessaan muokata kehitysjonoa sprinttien välillä riippuen siitä, miten työ etenee tai tarpeet muuttuvat. Scrummasterin tehtävänä on pitää huoli siitä, että kaikki tiimissä ymmärtävät scrumia ja käyttävät sitä oikein. Hän jakaa neuvoja kehitystiimille ja auttaa tuoteomistajaa ymmärtämään scrumin periaatteita. Kehitystiimi koostuu ammattilaisista, jotka toteuttavat ohjelmiston versiot. Kehitystiimin jäsenillä voi olla eri osaamisalueita, jotka kattavat kaikki projektin kehitystarpeet. (Schwaber ym. 2016, 5–6.)

### 3.3 Extreme programming

Extreme programming (XP) on yksi ketterien menetelmien lähestymistapa. Sen neljä tärkeintä arvoa ovat kommunikointi, yksinkertaisuus, palaute ja rohkeus. Lisäksi se sisältää neljä aktiviteettiä: ohjelmointi, testaus, kuunteleminen ja virheiden etsintä. Nämä arvot ja aktiviteetit johtavat XP:n kahteentoista käytäntöön. (Erickson, Lyytinen & Siau 2005, 90.)

XP:n käytännöt ovat:

- **Suunnittelupeli:** Jokaisen iteraation alussa asiakkaat, johtajat ja kehittäjät koontuvat määrittelemään ja arvioimaan seuraavan julkaisun vaatimukset.
- **Pienet julkaisut:** Kehitettävästä ohjelmistosta julkaistaan toimiva versio heti ensimmäisten iteraatioiden jälkeen. Uusia julkaisuja tehdään jatkuvasti.
- **Metaforat:** Ohjelmiston kuvaamiseen luodaan metafora.
- **Selkeä design:** Kehittäjät pyrkivät tekemään ohjelmistosta mahdollisimman yksinkertaisen.
- **Testaus:** Ohjelmointi aloitetaan testien kirjoittamisesta.
- **Refaktorointi:** Tuotetta kehitetään aina yksinkertaisempaan suuntaan.
- **Pariohjelmointi:** Kaksi kehittäjää työskentelee pareittain yhdellä tietokoneella.
- **Jatkuva integrointi:** Kehittäjät integroivat uutta koodia järjestelmään aina kun mahdollista.
- **Yhteisomistajuus:** Kaikki kehittäjät omistavat ohjelmakoodin ja voivat tehdä muutoksia, milloin haluavat.

- **Asiakas mukana työssä:** Asiakas on mukana kehitystyössä jatkuvasti ja vastaa kehittäjien kysymyksiin.
  - **40 tunnin viikot:** Työntekijöiden ei tarvitse tehdä ylitöitä.
  - **Avoin työskentelytila:** Tiimi työskentelee yhteisessä avoimessa työtilassa.
- (Cohen ym. 2003, 13.)

XP:ssä tuotetaan koodia sen mukaan, mitä vaatimuksia asiakas on määritellyt. Koodia testataan jatkuvasti, jotta voidaan varmistaa, että se toimii niin kuin kehittäjät sen suunnittelivat. Päämääränä on tuottaa toimiva tuote nopeasti. Tärkeimmät ominaisuudet toteutetaan ensiksi, ja uusia toiminallisuuksia voidaan tuoda lisää myöhemmässä vaiheessa. Kehittäjät keskittyvät yhteen iteraatioon kerrallaan eivätkä murehdi tulevista muutoksista. (Erickson ym. 2005, 91.)

#### 3.4 Ohjelmistokehitys Kisko Labsilla

Kisko Labsin ohjelmistokehitys noudattaa ketterän ohjelmistokehityksen menetelmiä. Kehitykselle ei ole asetettu mitään tiettyä menetelmää, vaan se mukautuu projektin tarpeiden tai asiakkaan toivomusten mukaan. Avainperiaatteita ovat nopeus, inkrementaalinen kehitys ja kommunikointi.

Scrum-viitekehityksen menetelmät ovat lähimpänä yrityksen toimintatapoja. Projekteilla on usein tuoteomistajana asiakkaan edustaja ja sprintit ovat yleensä viikon, joissain tapauksissa kahden, mittaisia. Jokainen tiimin jäsen kommunikoi tuoteomistajan kanssa ja osallistuu sprinttipalaveriin. Toiminnasta löytyy myös joitain XP:n piirteitä, kuten: pariohjelmointi, jatkuva integrointi ja avoin työskentelytila.

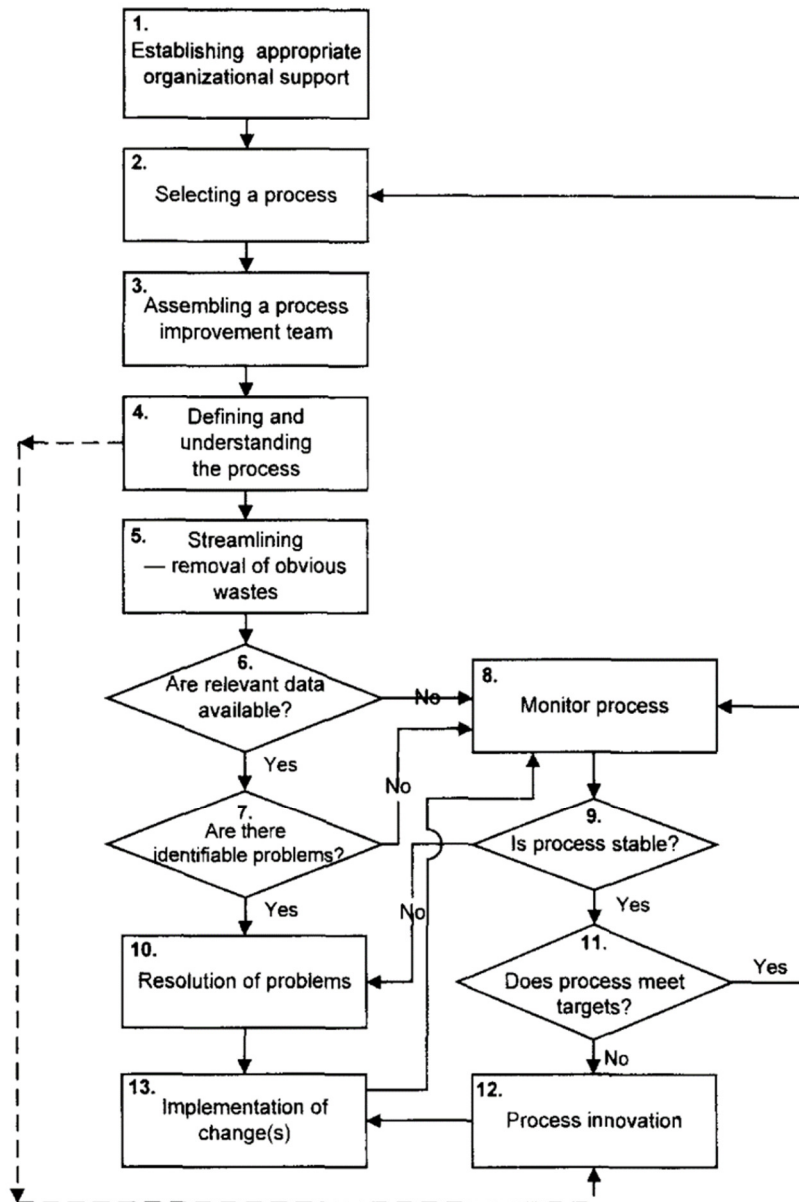
Yrityksen ohjelmistokehityksen tarkastelu on olennaista, sillä kehittäjät ja suunnittelijat kuuluvat samaan tiimin ja tekevät tiivistä yhteistyötä. Suunnittelutyö on ohjelmoinnin taapaa osa sprintteihin kuuluvaa työtä, ja suunnittelijat ovat osana koko kehitysprosessia. Suunnitteluprosessin ei saa heikentää toimivaa kehitystä, sen pitää täydentää sitä.

## 4 Prosessin määrittäminen

### 4.1 Aloitus

Prosessin määrittämisessä hyödynnettiin Rohlederin ja Silverin (1997, 139–154) *A tutorial on business process improvement*-ohjerunkoa. Ohje on luotu liiketoimintaprosessien mallintamiseen, mutta se soveltuu kaikenlaisiin prosesseihin. Ohjetta ei kehoiteta seuraamaan tarkasti, sillä luovuus on tärkeä osa prosessien parannusta. Ohjeessa ei olekaan kuvattu tarkempia toimenpiteitä, vaan se on kokoelma yleisluontoisia ohjenuoria, joiden perusteella voi kehittää yritykselle tai organisaatiolle sopivat keinot prosessin tarkempaan määrittelyyn.

Prosessin mallintaminen alkaa organisaation luottamuksen ja tuen saamisesta, sillä prosessien luominen ja käyttöönotto vaativat paljon resursseja. Seuraavaksi valitaan sopiva kehitettävä prosessi ja kehitystyöstä vastaavat henkilöt. Valittua prosessia muokataan ja parannetaan seuraavissa vaiheissa lopulliseen muotoon. *A tutorial on business process improvement* -artikkelin alkuperäinen kaavio määrittelyn etenemisestä esitettynä kuviossa 1.



Kuvio 1. Prosessin määrittämisen vaiheet. (Rohleder & Silver 1997, 141.)

Tässä tutkimuksessa organisaation luottamus ja kehitettävä prosessi oli määritetty heti alusta alkaen, sillä pyyntö suunnitteluprosessin määrittämiseen tuli Kisko Labsin johdolta. Suunnittelutyö oli toimivaa, mutta sille haluttiin silti määrittää tueksi prosessi. Prosessin määrittämiseen osallistuivat tämän työn kirjoittajan lisäksi firman muut suunnittelijat, mutta myös kehittäjät antoivat oman näkemyksensä. Prosessin määrittämiseen käytettiin haastattelututkimusta ja järjestettiin suunnittelutyöpajoja.

## 4.2 Haastattelututkimus

Koska aikaisempaa prosessia ei ollut, aloitettiin prosessin määrittäminen tutkimalla Kiskon työntekijöiden työskentelyä. Tietoa työskentelytavoista kerättiin haastattelututkimuksella. Haastattelu suunnattiin kaikille työntekijöille, suunnittelijoiden lisäksi myös kehittäjille. Kehittäjien näkökulman oletettiin antavan ulkopuolista näkökulmaa suunnittelutyöhön, ja muutenkin suunnittelutyön koettiin olevan erottamaton osa koko kehitysprosessia.

Haastattelu toteutettiin puolistrukturoituna haastatteluna. Puolistrukturoituun haastatteluun laaditaan valmiita kysymyksiä, jotka käydään järjestyksessä läpi. Kysymykset voivat olla täysin avoimia, jolloin saadaan aikaan vapaata keskustelua. Vastauksista voi tällöin selvittää asioita, joita kysymyksiä laatiessa ei voinut ennustaa. (Kurkela 2002.) Puolistrukturoitu haastattelu valittiin, jotta saataisiin aikaiseksi avointa ideointia prosessia varten.

Haastattelutilaisuus järjestettiin Kiskon toimistolla heinä- ja elokuun aikana 2015. Kysymyksiä laadittiin yhdeksän kappaletta, ja suurin osa niistä käsitteli nykyisten työskentelytapojen vahvuuksia ja heikkouksia. Lisäksi haastattelu sisälsi kysymyksiä, joiden tarkoituksena oli kartoittaa toivomuksia uudesta prosessista. Haastattelu pidettiin jokaiselle yrityksen senhetkisellem suunnittelijalle (lukuun ottamatta haastattelijaa), sekä kaikille kehittäjille. Haastattelutilaisuudessa oli kerralla vain haastattelija ja haastateltava. Haastattelun keskimääräinen pituus oli 36 minuuttia, ja jokaisessa syntyi avointa keskustelua.

Haastatteluista kerättiin muistiinpanot ja ne myös nauhoitettiin puhelimen avulla. Haastattelun jälkeen nauhoitukset kuunneltiin kertaalleen läpi ja muistiinpanoja täydennettiin tarkempien havaintojen pohjalta. Muistiinpanot lajiteltiin ja niistä etsittiin uusia ideoita ja parannuksia. Muistiinpanoista kerättiin yhtenäinen raportti, jossa esitettiin pahimmat ongelmakohdat nykyisissä työskentelytavoissa ja tärkeimmät toiveet uuden suunnitteluprosessin määrittämiseen.

#### 4.3 Haastattelun tulokset

Haastattelun ensimmäinen kysymyksen päämääränä oli selvittää haastateltavien oma näkemys nykyisen suunnittelutyön etenemisestä projekteissa. Suunnittelutyö oli pääpiirteittäin kaikkien vastaajien kesken samankaltainen, mutta poikkeamia löytyi ainakin projektin aloituksen vaiheista. Suurin osa haastateltavista olisi aloittanut suunnitteluprosessin asiakkaisiin tutustumisella ja heidän tarpeidensa ymmärtämisellä, mutta muutama haastateltavista olisi ryhtynyt heti piirtämään käyttöliittymäluonnoksia. Useimmat yrityksen kehittäjistä eivät osanneet jaotella vaiheita kuin muutamaan kohtaan: taustatutkimus, prototyypin kehittäminen ja iterointi. Suunnittelijat kuitenkin tunnistivat omasta työstään enemmän suunnitteluvaiheita, mikä auttoi prosessin kuvaamista myöhemmin. Tarkoitus ei kuitenkaan ole jakaa prosessia mahdollisimman pieniin osiin, mutta jonkinlainen vaiheistus auttaa kertomaan, miten projekti etenee, ja täten on helpompaa kommunikoida asiakkaalle työn etenemistä.

Seuraavat kysymykset käsittelivät nykyisen toimintamallin vahvuuksia, heikkouksia ja turhia työvaiheita. Joustavuus oli erittäin selkeä vahvuus kaikkien haastateltavien kesken; kukaan ei kaivannut prosessiin tarkkoja sääntöjä, sillä ne vain hankaloittaisivat työntekoa. Heikkouksista nousivat selkeästi esille kommunikointiongelmien, jota prosessin määrittämisessä lähdettiinkin alun perin ratkaisemaan. Toinen selkeä heikkous oli aikataulussa pysyminen johtuen siitä, ettei ole aina selkeää käsitystä, missä vaiheessa projekti on menossa ja kuinka paljon työtä on jäljellä. Aikatauluongelmaan palattiin, kun pohdittiin turhia työvaiheita; mikään ei tuntunut kokonaan turhalta, mutta joihinkin asioihin käytettiin liikaa aikaa. Liikaa aikaa haastateltavien mielestä kului käyttöliittymän rautalankamallien piirtämiseen, ja koettiin että HTML-prototyyppi on tehokkaampi ajankäytön kannalta. Vaikka joustavuus nähtiin valttina, pientä struktuuria kaivattiin työskentelellyn. Tämä voisi haastateltavien mielestä ratkaista aikatauluongelmia.

Haastateltavilta kysyttiin suunnittelutyön toimivuudesta suunnittelijoiden ja kehittäjien välillä. Yleinen mielipide oli, että yhteistyö toimii erittäin hyvin ja vain pieniä parannuksia tarvittaisiin kommunikointiin kehittäjien ja suunnittelijoiden välillä projektin alkuvaiheessa. Yhdessä työskenteleminen nähtiin positiivisena asiana ja sitä haluttiin lisätä entisestään. Seuraavaksi kysyttiin tyytyväisyyttä suunnittelutyön laatuun. Suurin osa piti kehitettyjä tuotteita laadukkaina ja oli saanut asiakkailta hyvää palautetta. Joissakin tapauksissa suunnittelutyön laatu oli kärsinyt pienestä budjetista ja resurssit oli pitänyt kes-

kittää ohjelmointityöhön. Viimeiseksi haastattelijoita pyydettiin kuvaamaan heidän ideaaliprosessinsa tai keksimään siihen ideoita. Moni oli tyytyväinen senhetkiseen työskentelyyn eikä halunnut muuttaa omaa työskentelyään radikaalisti. Päällimmäisenä toiveena oli lisätä työhön struktuuria prosessin avulla, ilman että työskentelyn joustavuus kärsisi. Suunnittelutyön haluttiin myös olevan enemmän iteratiivista. Iteraatioiden pitäisi olla pienempiä ja jokaisen iteraation jälkeen kysyttäisiin palautetta asiakkaalta. Haastattelukysymysten pohja on liitteenä (Liite 1).

#### 4.4 Haastattelun tulosten analysointi

Haastateltavat eivät osanneet vastata kysymyksiin ilman johdattelua. Kaikissa kysymyksissä käsiteltiin tyypillistä projektia, mutta kaikkien mielestä projektit ovat hyvin erityyppisiä. Tämä huolimatta siitä, että lähes kaikki projektit tehdään samalla teknologialla. Tämä vahvisti tarvetta joustavasta prosessista.

Prosessia ei haluttu määrittää lineaariseksi kaavioksi, sillä siitä joutuisi poikkeamaan lähes joka kerta. Seuraava idea oli tehdä prosessiin haarapolkuja, mutta jos jokaisesta vaiheesta lähtisi lukuisia haaroja, sen hyöty katoaisi. Lisäksi prosessista pitäisi ohittaa tai poistaa vaiheita, jos ne eivät ole relevantteja projektin kannalta. Projekti voi olla esimerkiksi vain visuaalista suunnittelua, jolloin siitä pitäisi poistaa kaikki perinteisemmät vaiheet, kuten käyttöliittymäsuunnittelu tai prototyypin rakentaminen.

Yksimielinen mielipide oli, ettei perinteinen lineaarinen prosessimalli sovi yrityksen toimintaan eikä kulttuuriin. Prosessista haluttiin joustava, ja keskusteluissa syntyi idea projektikohtaisesti räätälöidystä prosessista. Prosessin rakentamisen tueksi kerättäisiin koelma kaikista mahdollisista työvaiheista ja näistä vaiheista valittaisiin sopivimmat kuhunkin projektiin. Tällä tavalla jokaisen projektin alussa (tai jo tarjousvaiheessa) päätettäisiin tarvittavat suunnittelukeinot. Tämä idea sai kannatusta, koska se ei muuttaisi senhetkisiä työskentelytapoja mutta pakottaisi miettimään jokaisen projektin prosessin struktuuria. Uuden prosessin tarkoitus ei alun perinkään ollut korvata vanhoja työskentelytapoja vaan tuoda ne esille prosessin muodossa. Riskiksi tässä lähestymistavassa havaittiin liikakin joustavuus, mutta vähäinenkin struktuuri on parempi kuin ei mitään.



#### 4.5 Ensimmäinen työpaja

Haastatteluiden analysoinnin jälkeen sovittiin järjestettäväksi työpaja suunnittelijoiden kesken. Työpaja järjestettiin maaliskuussa 2016 ja se kesti kaksi tuntia. Siihen osallistui yhteensä neljä Kiskon suunnittelijaa, mukaan lukien tämän työn kirjoittaja. Työpajan tavoitteena oli hahmotella uutta suunnitteluprosessia.

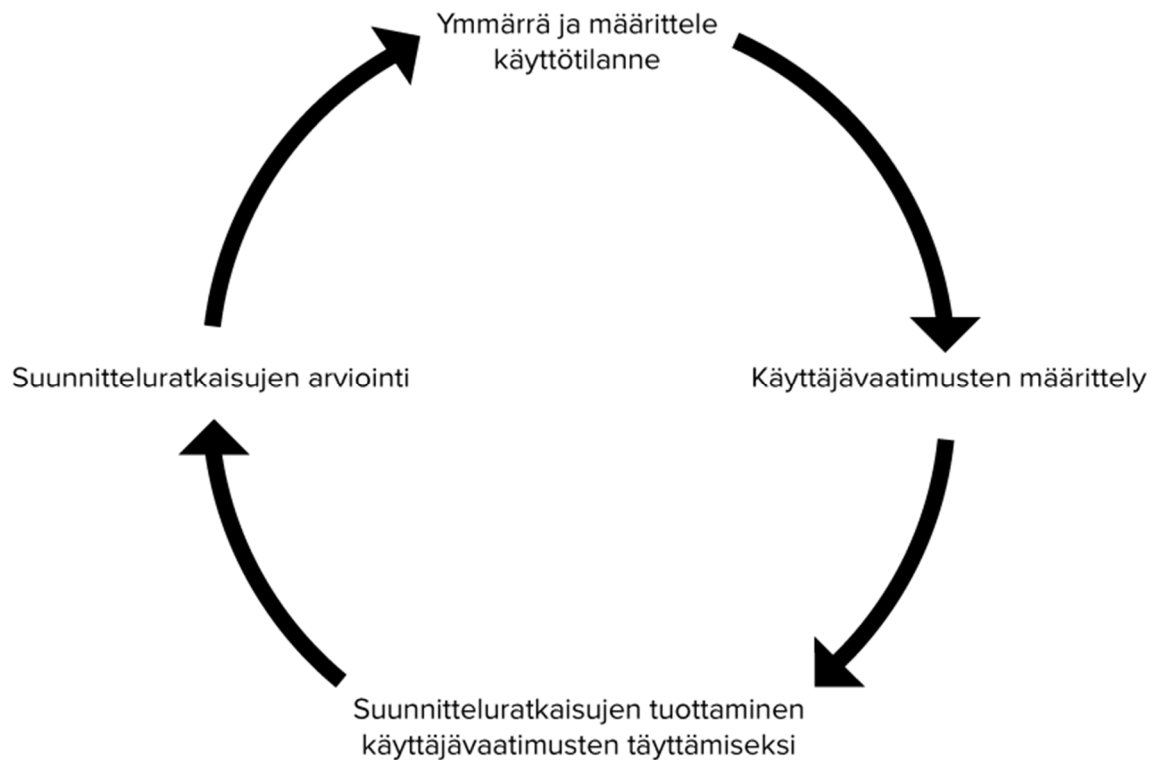
Ensiksi työpajassa käytiin läpi haastatteluiden tulokset, joita varten oli laadittu selkeä esitys. Esityksessä jokaisen kysymyksen tulokset oli koottu yhteen ja vastauksista oli poimittu tärkeimmät havainnot. Havainnoista keskusteltiin yhdessä ja johtopäätöksistä laadittiin muistiinpanot. Esityksen loppuun oli laadittu ideoita suunnitteluprosessin kuvausta varten. Nämä kuvaukset käytiin läpi ja keskusteltiin, mikä olisi toimivin ratkaisu Kiskolle.

Haastattelun tulosten perusteella syntynyt idea projektikohtaisesti räätälöidystä prosessista sai hyväksynnän, mutta se ei vielä tuntunut tarpeeksi selkeältä prossilta. Työpajassa ehdotettiin prosessin jakamista korkean tason prosessiin ja alemman tason aktiviteetteihin. Korkean tason prosessi kuvaisi yleisesti suunnittelutyön periaatteita ja alempi taso sitoutuisi korkeamman tason vaiheisiin. Työpajassa esitettiin kaksi ehdotusta korkean tason mallin kuvaamiseksi: käyttäjäkeskeinen suunnitteluprosessi ja tuplatimantti-suunnitteluprosessi.

Käyttäjäkeskeisessä suunnitteluprosessissa on neljä vaihetta:

1. Käyttötilanteen ymmärtäminen ja määrittely
2. Käyttäjävaatimusten määrittely
3. Suunnitteluratkaisujen tuottaminen
4. Suunnitteluratkaisujen arviointi

Prosessi on iteratiivinen, jokainen vaihe on riippuvainen edellisen vaiheen tuotoksista. (Suomen standardisoimisliitto SFS 2010, 18-28.) Käyttäjäkeskeinen suunnitteluprosessi on kuvattu kuviossa 2.

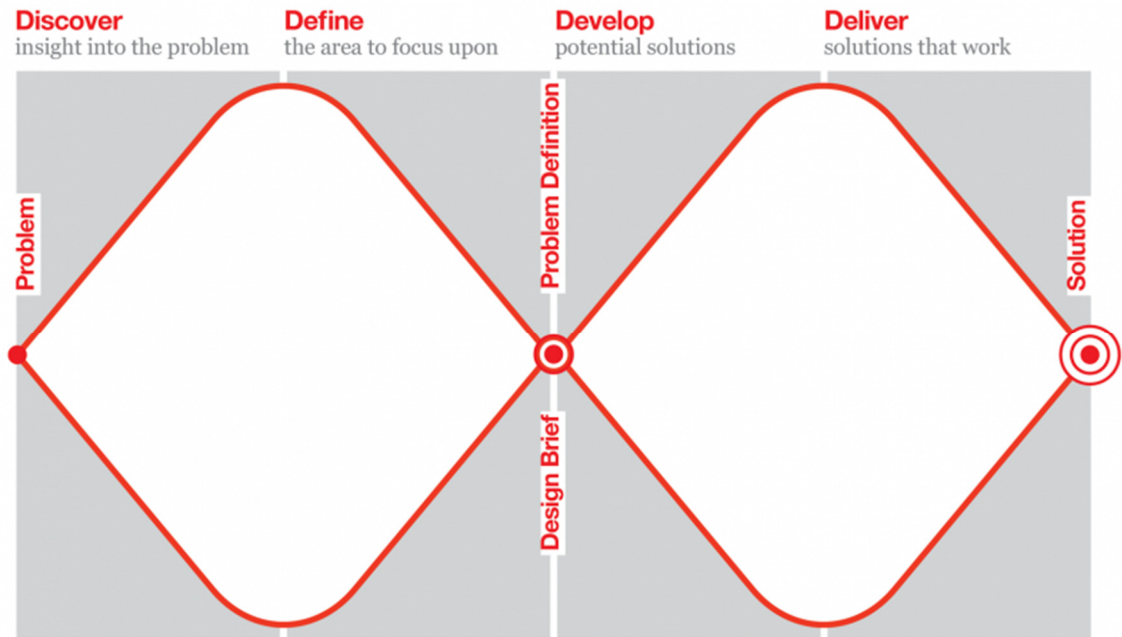


Kuvio 2. Käyttäjäkeskeinen suunnitteluprosessi. Piirretty perustuen Standardiin: SFS-EN ISO 9241-210 (Suomen standardisoimisliitto SFS 2010, 28).

Tuplatimantti-prosessissa on neljä vaihetta:

1. Löydä
2. Määrittele
3. Kehitä
4. Toimita

Ensimmäinen vaihe kuvaa projektin alkua, jossa suunnittelijat keräävät useita mahdollisia ideoita suunnitteluongelman ratkaisemiksi. Toisessa vaiheessa näistä ideoista määritetään paras ratkaisu valittuun ongelmaan. Seuraavassa vaiheessa iteroidaan eri toteutuksia esimerkiksi prototyyppien avulla. Viimeisessä vaiheessa toimitetaan lopullinen toimiva ratkaisu. (British Design Council 2005.) Tuplatimantti-prosessi on kuvattu kuviossa 3.



Kuvio 3. Tuplatimantti-suunnitteluprosessi. (British Design Council 2005)

Työpajan keskustelussa kummankin mallin vaiheet koettiin samankaltaisiksi, mutta tuplatimantissa ne oli nimetty yleisluontoisemmin. Prosessien visuaalinen esitys oli ratkaiseva tekijä valinnassa. Käyttäjäkeskeisen suunnitteluprosessin syklinen malli oli tutumpi, mutta myös tylsempi. Sen myös koettiin sekoittuvan liian helposti ketterän kehityksen iteratiiviseen malliin ja suunnitteluun haluttiin erottuvan kehityksestä. Lopulta päädyttiin yksimieliseen ratkaisuun käyttää tuplatimanttimalia.

Korkeamman tason prosessimalliksi valittiin tuplatimantti, johon alemman tason aktiviteetit sijoittuisivat projektista riippuen. Alemman tason aktiviteetteja ei oltu vielä määritetty ja ilman niitä prosessin käyttöönotto ei olisi mahdollista. Suunnittelijoiden kesken sovittiin järjestettäväksi toinen työpaja, jonka agendana olisi näiden aktiviteettien tarkempi määrittäminen ja sijoittaminen tuplatimantin vaiheisiin.

#### 4.6 Toinen työpaja

Toiseen työpajaan osallistui kolme suunnittelijaa. Työpaja järjestettiin syyskuussa 2016. Kokouksessa mietittiin, mitä eri suunnittelutyöhön liittyviä työtehtäviä eri projekteissa tehdään tai on aikaisemmin tehty. Suunnittelijat keksivät vuorotellen ideoita ja ne kirjattiin

muistiinpanoihin. Tehtävät lajiteltiin aktiviteeteiksi tuplatimantin neljään eri vaiheeseen. Tuloksen syntyä seuraava alustava lista:

Löydä	
Aktiviteetti	Kuvaus
Kickoff-työpaja	Yhteinen työpaja asiakkaan edustajien kanssa, jossa on tarkoituksena selvittää tarkemmin asiakkaan tarpeet. Työpaja voi sisältää yhdessä tehtäviä harjoituksia, kuten unelmakäyttöliittymän piirtämistä tai tärkeimpien ominaisuuksien listausta.
Konseptointi-työpaja	Työpaja jossa alkuperäistä ideaa jalostetaan pidemmälle. Pyritään vastaamaan kysymyksiin "Mitä ongelmaa ratkaistaan?" ja "Miten ongelma ratkaistaan?".
Ideointi- ja suunnittelutyöpaja	Tähän vaiheeseen voi osallistua useampi suunnittelija ja asiakkaan edustaja. Tarkoituksena on suunnitella yhdessä yksityiskohtaisemmin palvelun ominaisuuksia, ja hahmotella jo ensimmäisiä käyttöliittymäratkaisuja.
Käyttäjätutkimus	Kartoitetaan uusia tai olemassa olevia käyttäjiä. Tutkimuksessa voidaan laatia persoonia tai analysoida olemassa olevaa dataa käyttäjistä. Persoonat ovat kuvauksia tyypillisimmistä käyttäjistä ja niiden pitää perustua aina oikeisiin havaintoihin.
Käyttäjähaastattelu	Toteutetaan haastattelututkimus potentiaalisille käyttäjille ja varmennetaan heidän tarpeensa.
Teknologiakartoitus	Tutkitaan yhdessä kehittäjien kanssa vaadittavia teknologioita. Teknologia voi rajoittaa suunnitteluratkaisuja varsinkin, jos resurssit ovat rajalliset.
Kilpailija-analyysi	Havainnoidaan muiden samankaltaisten palveluiden toteutuksia. Vaikka kehitettävä palvelu olisi täysin uudenlainen, voi kilpailijoiden palveluista löytyä ideoita johonkin tiettyyn yksityiskohtaan.

Määrittele	
Aktiviteetti	Kuvaus
Käyttäjätarinat	Kuvataan mitä käyttäjä tekee uudella palvelulla ja miten se muuttaisi hänen käyttäytymistään. Työskentelyyn suunnatussa työkalussa voidaan tarkastella, miten nykyinen työprosessi muuttuisi uuden työkalun myötä.
Käyttäjäpolun suunnittelu	Mietitään kaikki mahdolliset toiminnot mitä käyttäjä voi tehdä. Kuvataan tarkasti mitä linkkejä käyttäjä joutuu painamaan suorittaakseen tietyn toiminnon.
Rautalankamallien piirtäminen	Piirretään karkea rautalankamalli käyttöliittymän näkymistä. Työkaluina usein paperi ja kynä, mutta mahdollisuus käyttää myös piirto-ohjelmia tietokoneella.

Informaatioarkkitehtuurin suunnittelu	Suunnitellaan palvelun hierarkia, eli mikä linkki johtaa mihinkin näkymään. Kaikki linkit ja näkymät nimetään selkeästi ja ymmärrettävästi.
Styleguiden suunnittelu	Toteutetaan styleguide, joka määrittää palvelun käyttöliittymän ulkoasua. Styleguide voi perustua asiakkaan brändiin tai ulkoasu kehitetään varta vasten uuden palvelun tueksi. Styleguide rakennetaan yleensä HTML:llä ja CSS:llä, jotta sitä voidaan suoraan hyödyntää lopullista käyttöliittymää rakennettaessa.
Interaktiiviset rautalankamallit	Toteutetaan tietokoneella tai puhelimella toimiva interaktiivinen rautalankamalli käyttöliittymästä, jota käyttäjät voivat testata. Toteutus esimerkiksi InVision-työkalun avulla ( <a href="http://www.invisionapp.com">www.invisionapp.com</a> ).

Kehitä	
Aktiviteetti	Kuvaus
HTML-prototyyppi	Toteutetaan selaimessa toimiva prototyyppi käyttöliittymästä. Prototyypin koodia käytetään suoraan lopullisessa toteutuksessa, siksi se on osa kehitysvaihetta. HTML-prototyyppi mahdollistaa käyttöliittymän testauksen sen lopullisessa ympäristössä selaimessa. Ero lopulliseen käyttöliittymään on staattinen sisältö, sillä prototyyppi ei käytä tietokantaa.
Infografiikka / Datan visualisointi	Datan visualisointi voi joissakin projekteissa olla tärkeä osa toteutusta. Visualisoinnin toteutus on jaettu erilliseksi aktiviteetiksi, sillä se vaatii huomattavan määrän perehtymistä datan analysointiin. Kuvaajien tekninen toteutus vaatii usein myös paljon resursseja.
Verkkosovelluksen käyttöliittymän kehitys	Suunnittelijat vastaavat käyttöliittymien teknisestä toteutuksesta. Selaimella toimivien palveluiden toteutus tehdään HTML, CSS ja Javascript -kieliä käyttäen.
Hybridi mobiilisovelluksen käyttöliittymän kehitys	Hybridi mobiilisovellus on puhelimella toimiva sovellus, jonka sisällä toimii responsiivinen verkkosovellus. Suunnittelija vastaa verkkosovelluksen käyttöliittymän toteutuksesta ja varmistaa sen toimivuuden mobiiliapplikaation sisällä.
Natiivi mobiilisovelluksen käyttöliittymän kehitys	Natiivi mobiilisovellus toimii täysin itsenäisesti jokaisella alustalla. Käyttöliittymän suunnittelee suunnittelija, mutta toteutuksen tekee sovelluskehittäjä. Tämä aktiviteetti on ainoa poikkeus, missä suunnittelija ei vastaa teknisestä toteutuksesta.
Brändisuunnittelu	Brändisuunnittelussa luodaan brändiohjeistus asiakkaalle. Brändi luodaan yleensä kehitettävää palvelua varten, mutta siitä voidaan johtaa laajempi ohjeistus kattamaan myös muut tarpeet. Ohjeistoon voi kuulua esimerkiksi värit, typografia, logo ja printtimateriaalia.
Ikonien suunnittelu	Palveluun luodaan omaperäisempi ilme piirtämällä ikoneita. Usein käytetään valmiita verkosta saatavia ikoneita, mutta jos

	ulkoasusta halutaan yhtenäisempi brändin kanssa, voidaan ikonit piirtää itse.
--	---

Toimita	
Aktiviteetti	Kuvaus
Työn loppuun hiominen	Tässä vaiheessa käyttöliittymä on jo täysin toimiva ja voidaan käyttää aikaa pieniin parannuksiin. Parannukset vaikuttavat lähinnä ulkoasuun, mutta myös pieniä käytettävyyssparannuksia voidaan tehdä. Asiakkalle on kommunikoidava, että käyttöliittymän visuaalinen ulkoasu määräytyy vasta ihan lopuksi.
Testaus	Valmista palvelua testataan ja varmistetaan, että kaikki ominaisuudet toimivat. Testausta tehdään kokoajan, mutta lopuksi tehdään kattavampi testaus, johon kuuluu esimerkiksi selain-testaus.
Iterointi	Ei varsinaisesti yksittäinen aktiviteetti. Kuvaakin vaiheiden iteratiivista etenemistä. Viimeiseen vaiheeseen päästään iteroinnin kautta.
Julkaisu	Palvelu julkaistaan loppukäyttäjille ja palautetta otetaan vastaan. Mahdolliset ongelmat ovat yleensä teknisiä, mutta suunnittelija voi myös saada kiireellisiä korjaustehtäviä.
Markkinointi	Projekteissa keskitytään yleensä itse palvelun kehittämiseen, mutta joskus sen markkinointiin tarvitaan suunnittelijan apua. Suunnittelija voi luoda esimerkiksi erillisen markkinointisivun, jossa uusi palvelu esitellään.
Jatkokehitys	Julkaisun jälkeen pohditaan jatkokehitystarpeita. Jatkokehitys voi olla pienten ominaisuuksien lisäämistä tai isompien kokonaisuuksien kehittämistä, joita varten tarvitaan kokonaan uusi projekti.

Ensimmäiseen vaiheeseen tuotiin kaikki asiakkaan ja käyttäjän tarpeiden kartoitukseen liittyvä tutkimustyö. Määrittelyvaiheessa on suunnittelutyötä, joka toimii apuna konseptin hahmottamisessa ja tukee varsinaista kehitystyötä. Kehitysvaiheessa on lopputuotteen toteutukseen suoraan vaikuttavia työvaiheita. Viimeiseen vaiheeseen ei jäänyt mitään konkreettisia suunnittelutehtäviä, koska kehitysvaiheen tehtävien katsotaan olevan iteratiivisia. Kehitysvaiheiden iteratiivinen toteutus johtaa väistämättä tuotteen valmistumiseen ja toimitukseen.

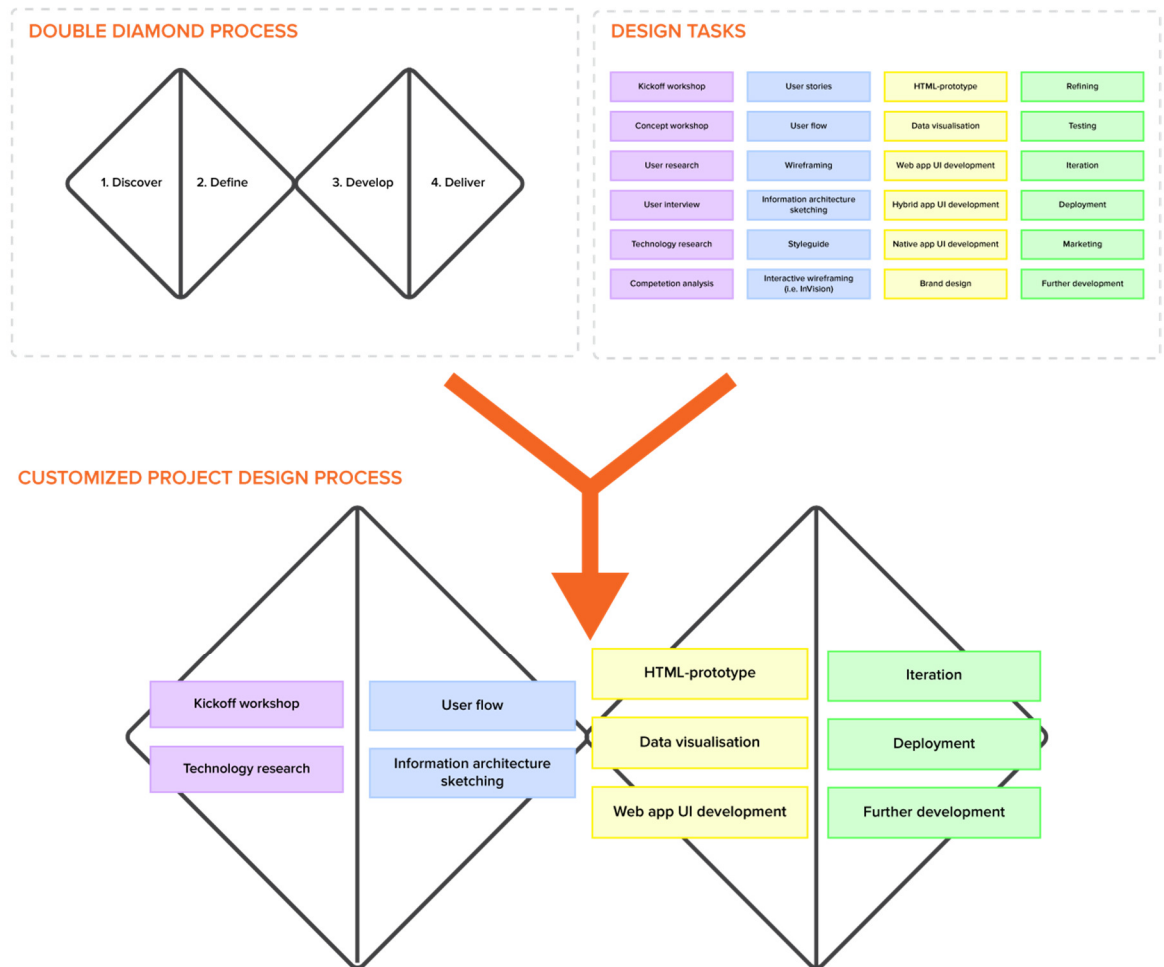
## 5 Johtopäätökset

### 5.1 Suunnitteluprosessin kuvaus

Suunnitteluprosessin kuvaus syntyi täysin sisäisesti Kisko Labsin työntekijöiden toimesta. Yrityksen kulttuuriin kuuluu, että työntekijät saavat vaikuttaa omaan työskenteelyynsä ja työympäristöönsä. Prosessiin koitettiin hakea ideoita valmiista suunnitteluprosesseista, mutta mikään ei sopinut suoraan olemassa oleviin käytäntöihin. Suunnittelu on käsitteenä todella laaja ja useat olemassa olevat prosessit oli alun perin kehitetty esimerkiksi teolliseen muotoiluun ja niiden muokkaaminen ohjelmistoalalle tuntui epäluontevalta. Kisko on myös pienenä yrityksenä toimintatavoiltaan hieman erilainen, joten persoonallisen ja omalaatuisen prosessin määrittäminen tuntui luontevalta.

Prosessin jakaminen kahteen tasoon onnistuu kommunikoimaan suunnittelutyötä myös kahdella tasolla. Korkeampi taso kuvaa yleisesti suunnittelutyön luonnetta ongelman ratkaisuna, eikä pelkästään visuaalisena suunnitteluna. Käyttökokemussuunnittelu on tärkeä osa verkkopalveluiden kehitystä ja sen tuoman arvon esittäminen on ollut aina ongelmallista. Alempi taso taas kuvaa konkreettista työtä ja se kertoo, mihin suunnittelijan työaika kuluu.

Prosessi on myös todella joustava, sillä korkeampi taso ei ota tarkemmin kantaa suunnittelutyöhön ja alempi taso on täysin räätälöitävissä tarpeiden mukaan. Kuviossa 4 on esitetty esimerkki, miten projektin prosessi rakennetaan.



Kuvio 4. Esimerkki prosessin kustomoinnista projektin tarpeiden mukaan.

Tuplatimantti-mallin ainoa ongelma on iteratiivisen suunnittelun hahmottaminen. Timantin on tarkoitus kuvastaa iteratiivista lähestymistapaa, kun lähdetään timantin reunasta ja etsitään ratkaisuja laajentuen timantin huippuihin. Tämä on kuitenkin osoittautunut vaikeaksi ymmärtää ilman selitystä. Ideana on laajentaa timantin kuvausta sisältämään ehkä syklinen rakenne (pyörivät nuolet).

## 5.2 Integroituminen kehitysprosessiin

Vaikka suunnittelulle on määritetty oma prosessi, se ei tarkoita, että se olisi ketterän kehitysprosessin ulkopuolella. Timantit kuvaavat koko projektin etenemistä ja alemman tason aktiviteetit voidaan helposti sisällyttää ketterän kehityksen sprintteihin. Kehitys- ja suunnittelutyö jatkavat vanhoilla periaatteilla, mutta suunnittelu saa omasta prosessista



hiukan enemmän struktuuria. Voidaan ajatella, että ketterä kehitys on koko yrityksen pääprosessi ja suunnitteluprosessi on sen alaprosessi.

Suunnittelu käyttää oman prosessin tukena myös ketterän kehityksen menetelmiä. Ketterät menetelmät auttavat suunnittelijaa samaan jatkuvaa palautetta ja iteroimaan työtään nopeammin. Näin vältetään uudelleensuunnittelulta, jos asiakas ei pidäkään lopputuloksesta. Ketterien menetelmien käyttäminen helpottaa yhteistyötä kehittäjien kanssa, jotka käyttävät samaa prosessia omassa työskentelyssään. (Clum 2013.) Työ tehdään pieninä osina ja asiakkaalta tai käyttäjiltä pyritään kysymään palautetta jatkuvasti. Asiakas voidaan myös ottaa mukaan suunnittelutyöpajoihin, jolloin ymmärretään tarpeet paremmin. Pariohjelmoinnin vastikkeeksi on kokeiltu myös parisuunnittelua, joka on toiminut loistavasti.

Ketteriä menetelmiä voidaan myös käyttää itsenäisesti suunnittelutyössä. *Design sprint* -kirjassa kerrotaan keinoista, joilla yritykset voivat nopeuttaa suunnitteluprosessiaan. Kirjan mukaan design sprint -käytäntöjä voidaan hyödyntää yksittäisiin suunnitteluongelmiin missä vaiheessa projektia tahansa. (Banfield, Lombardo & Wax 2015.) Keino muistuttaa hyvin paljon ketteriä menetelmiä, sillä siinä tärkeimmässä roolissa on nopea iterointi. Kirjan mukaan menetelmä on suunnattu nimenomaan digitaalisten tuotteiden suunnittelulle ja integroituu ketterien menetelmien tavoin hyvin kehitysprosessiin. (Banfield ym. 2015) Design sprint ei kuitenkaan ole kokonainen prosessi vaan pikemminkin miniprosessi, jota voi soveltaa pieniin yksittäisiin vaiheisiin.

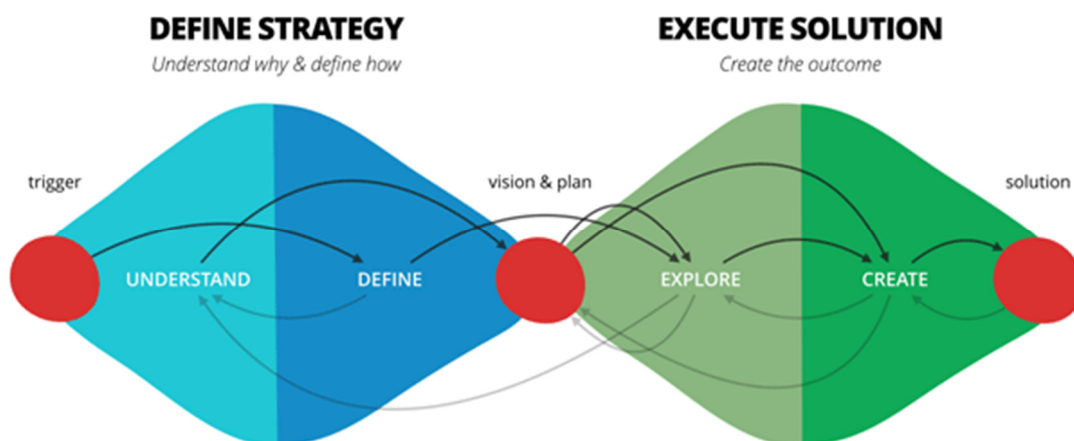
Kiskon erikoisuus on myös suunnittelijoiden tekniset taidot. Muissa yrityksissä Kiskon suunnittelijat olisivat pikemminkin front-end-kehittäjiä, mutta Kiskolla käyttöliittymän tekeminen toteutus on osa suunnittelijan työtä. Jos sama henkilö suunnittelee ja toteuttaa palvelun alusta loppuun, voidaan välttyä kommunikaatio-ongelmilta. Tämänkin takia on loogista, että kehitys- ja suunnitteluprosessit ovat tiiviisti yhdessä.

### 5.3 Jatkotoimenpiteet

Prosessikuvausta ei ole virallisesti otettu käyttöön. Suunnittelijat tietävät prosessin, mutta se ei vaikuta merkittävästi olemassa oleviin työskentelytapoihin. Se on kuitenkin auttanut suunnittelijoiden oman työn vaiheistamisessa ja auttanut karsimaan turhia työskentelyvaiheita. Esimerkiksi prototyyppien kehittämiseen on valittu useammassa tapauksessa tehokkaammat työtavat siirtymällä täysin HTML-prototyyppien kehittämiseen.

Asiakkaille prosessia ei ole vielä kokonaisuudessaan esitelty. Tarjouksissa suunnittelu-työtä on jaoteltu tarkemmin alemman tason aktiviteettien perusteella, mutta tuplatimantti-ia ei ole vielä käytetty. Luontevin tilanne prosessin esittelylle olisi projektin aloituspalaverissa, jossa käydään muutenkin käytännön asioita ja aikataulua läpi. Prosessi voitaisiin esittää pitempiaikaiselle asiakkaalle, jolloin saataisiin palautetta prosessin parannuksista verrattuna aikaisempaan tilanteeseen.

Prosessin kuvaaminen vaatii myös jatkokehitystä. Tuplatimantti-mallia voisi muokata sopivammaksi ohjelmistokehitysympäristöön. Kuviossa 5 esimerkki muokatusta tuplatimantista. Alemman tason aktiviteetit vaativat myös lisäyksiä ja muutoksia. Monet tehtävät ovat liian laajoja (esim. verkkosovelluksen käyttöliittymän kehitys) ja niitä voisi pilkkoa pienemmiksi kokonaisuuksiksi.



Kuvio 5. Ohjelmistokehitykseen muokattu tuplatimantti-prosessi (Schneider 2015).

Isoin muutos toimintatapoihin on projektin prosessin räätälöinti. Jokaisen projektin alussa pitäisi olla tavat ja keinot luoda projektille oma prosessi. Toki jokaisessa projektissa mietitään, mitä suunnittelukeinoja tarvitaan, mutta prosessin luominen vaatii hiukan enemmän panostusta. Tätä varten voisi kehittää työkaluja helpottamaan kaavioiden luomista. Yksinkertaisin ratkaisu on piirtää tuplatimantti-malli isolle julisteelle ja kirjoittaa kaikki alemman tason aktiviteetit muistilapuille. Muistilaput voisi siirtää tuplatimantin päälle oikeisiin vaiheisiin ja ottaa kuva valmiista prosessikuvauksesta. Toinen ratkaisu on kehittää web-selaimessa toimiva työkalu, jossa aktiviteetit voidaan raahata hiirellä

oikeisiin kohtiin. Digitaalinen työkalu helpottaisi prosessikaavion jakamista asiakkaalle, mutta kokemuksena fyysinen ratkaisu voi olla mukavampi.

Rohlederin ja Silverin (1997) ohjeistuksen yksi vaiheista on prosessin monitoroiminen ja ongelmien havaitseminen ja niihin reagoiminen. Monitorointia voidaan tehdä vasta kun prosessi on aktiivisemmassa käytössä. Kiskolla työn tehokkuutta ja mielekkyyttä parannetaan muutenkin aktiivisesti. Tämän prosessin monitoroinnissa tärkeätä on saada palautetta ulkopuolelta ja havaita potentiaalisia ongelmia asiakkaan kautta. Monitoroinnin lisäksi prosessin hyötyjen mittaus on yksi mahdollisuus saada lisäarvoa prosessista. Jos tulevat projektit käyttävät samaa prosessimallia, on niitä helpompi vertailla keskenään. Vertailu voi olla hankalaa, johtuen prosessin joustavuudesta, mutta jokaisesta prosessista löytyy varmasti yhtäläisyyksiä. Prosessin vaiheiden mittaamisella voitaisiin saada dataa työvaiheiden kestosta, onnistumisesta ja yleisyydestä.

## 6 Yhteenveto

Opinnäytetyön tavoitteena oli määritellä suunnitteluprosessi Kisko Labs Oy:lle. Aikaisempaa prosessikuvausta ei ollut, joten prosessi rakennettiin olemassa olevien työskentelytapojen mukaan ja se pidettiin osana ketterää kehitystyötä. Tuloksena ei syntynyt täysin uutta prosessia, vaan prosessikuvaus jo toimivasta suunnittelutyöstä. Suunnitteluprosessi on todella kevyt ja joustava, eikä prosessikuvauksen oleteta juurikaan muuttavan työskentelyä. Vaikka prosessikuvaus ei vaikuta päivittäiseen työskentelyyn, oli sen määrittämisestä paljon hyötyä. Määrittelyn aikana tehty haastattelututkimus ja työpajat herättivät keskustelua työtapojen toimivuudesta. Oman työskentelyn tarkkailu auttoi tuomaan esille mahdollisia ongelmakohtia, ja suunnittelutyöstä löydettiin parannuskohteita.

Prosessikuvausta ei ole vielä otettu kokonaisuudessaan käyttöön, mutta sen periaatteet ovat selvillä työntekijöille. Suunnitteluprosessin etenemiseen on kiinnitetty enemmän huomiota ja sitä pyritään parantamaan jatkuvasti. Suunnitteluprosessin kaksi tasoa kommunikoivat onnistuneesti suunnittelutyön luonnetta. Korkeamman tason tuplatimanttimalli kuvaa hyvin suunnittelutyön ongelmanratkaisua ja sopii täten hyvin Kiskon viestiin ”Autamme löytämään oikean tien”. Alemman tason aktiviteetit kuvaavat konkreettista työtä ja ne kertovat hyvin, mitä suunnittelija päivittäin oikeasti tekee.

Yksi prosessin tavoitteista oli parantaa asiakaskommunikaatiota, mutta sen testaaminen jäi tämän opinnäytetyön ulkopuolelle, koska prosessikuvausta ei ole vielä esitetty Kiskon asiakkaille. Suunnittelijat osaavat nyt kuitenkin paremmin kertoa työstään ja he voivat kommunikoida asiakkaille paremmin työn etenemistä. Tarkoitus on kuitenkin esitellä prosessi asiakkaille tulevaisuudessa, minkä toivotaan luovan lisää luottamusta suunnittelu-työtä kohtaan.

Opinnäytetyön teoria käsitteli paljon ohjelmistokehityksen menetelmiä, mutta ei valmiita suunnitteluprosesseja. Tämä oli tietoinen valinta, koska tiedettiin, että yritys haluaa omanlaisensa suunnitteluprosessin. Olisi kuitenkin ollut hyvä tehdä lyhyt katsaus eri suunnitteluprosesseihin ja -menetelmiin. Näin olisi voitu verrata Kiskon suunnitteluprosessia muihin, samaan tapaan kuin ohjelmistokehitysmenetelmiä verrattiin yleisimpiin ketteriin menetelmiin kolmannessa luvussa.

## Lähteet

Abrahamsson, Pekka & Salo, Outi & Ronkainen, Jussi & Warsta, Juhani 2002. Agile software development methods: Review and analysis. VTT Publications 478, Espoo: VTT.

Banfield, Richard & Lombardo, Todd & Wax, Trace 2015. Design Sprint. Sebastopol, Kalifornia: O'Reilly Media, Inc

Beck, Kent & Beedle, Mike & van Bennekum, Arie & Cockburn, Alistair & Cunningham, Ward & Fowler, Martin & Grenning, James & Highsmith, Jim & Hunt, Andrew & Jeffries, Ron & Kern, Jon & Marick, Brian & Martin, Robert & Mellor, Steve & Schwaber, Ken & Sutherland, Jeff & Thomas, Dave 2001. Manifesto for Agile Software Development. <<http://agilemanifesto.org/>> (Luettu 23.4.2017).

British Design Council 2005. The Design Process: What is the Double Diamond? <<http://www.designcouncil.org.uk/news-opinion/design-process-what-double-diamond>> (luettu 6.5.2017)

Cohen, David & Lindvall, Mikael & Costa, Patricia 2003. Agile software Development A DACS State-of-the-Art Report. Maryland: Fraunhofer Center.

Clum, Luke 2013. Applying Agile Principles to Design. <<http://webdesignledger.com/applying-agile-principles-to-design>> (Luettu 22.11.2015).

cPrime Inc 2017. What is agile? What is Scrum? <<https://www.cprime.com/resources/what-is-agile-what-is-scrum/>> (luettu 26.4.2017).

Duka, Denis 2013. Adoption of agile methodology in software development. Information & Communication Technology Electronics & Microelectronics (MIPRO), 2013 36th International Convention, sivut 426 – 430.

Erickson, John & Lyytinen, Kalle & Siau, Keng 2005. Agile Modeling, Agile Software Development, and Extreme Programming: The State of Research. Journal of Database Management 16 (4), sivut: 88 – 100

Kurkela, Reijo 2002. Haastattelutavat – Puolistrukturoitu haastattelu. <<https://www.stat.fi/virsta/tkeruu/04/02/>> (Luettu 14.5.2017).

Martinsuo, Miia. & Blomqvist, Marja 2010. Prosessien mallintaminen osana toiminnan kehittämistä. Opetusmoniste. Tampere: Tampereen teknillinen yliopisto. Teknis-taloudellinen tiedekunta.

Radigan, Dan 2017. The Agile Coach: Scrum. <<https://www.atlassian.com/agile/scrum>> (Luettu 26.4.2017).

Rohleder, Thomas. & Silver, Edward. 1997. A tutorial on business process improvement. Journal of Operations Management 15, sivut 139 – 154.

Schneider, Jonny 2015 The Double Diamond: Strategy + Execution of the Right Solution. <<https://www.thoughtworks.com/insights/blog/double-diamond>> (Luettu 7.5.2017).

Schwaber, Ken & Sutherland, Jeff 2016. The Scrum Guide. <<http://www.scrumguides.org/docs/scrumguide/v2016/2016-Scrum-Guide-US.pdf>> (Luettu 26.4.2017).

Sharma, Ashish 2013. Essential, Valuable, Timely Documentation. <<https://www.scrumalliance.org/community/articles/2013/december/essential-valuable-timely-documentation>> (Luettu 14.5.2017).

Standardi SFS-EN ISO 9241-210. 2010. Suomen standardisoimisliitto SFS.. Helsinki: Suomen standardisoimisliitto SFS.

Williams, Laurie & Cockburn, Alistair 2003. Agile software development: it's about feedback and change. Computer (Volume: 36, Issue: 6 2003 ), sivut 39 – 43.

## **Puolistrukturoidun haastattelun kysymykset**

Point of this interview is to survey employees' perception of the current design process. The application of the design process is the typical web project the company usually does.. Design in this context means basically tasks that are not programming so for example: User research, Information architecture, User interface design, Wireframing, Sketching, Prototyping, Visual design, Brand design etc.

You don't need to have an answer to every question it's assumed that the process is not perfectly clear to everyone and that's the reason we are trying to define it.

1. What are the first steps in the current design process when starting a new project?
2. Can you describe in detail how the current process works? Try to recall what steps a past project had.
3. What are the most crucial steps in the process?
4. Is there something missing from the process?
5. Are there any unnecessary steps in the process?
6. What problems or bottlenecks does the process have?
7. How well does the design process integrate with the development process? Is there enough communication between designers and developers?
8. Are you personally satisfied with the design output (deliverables)? What about clients? If not, why?
9. What would your ideal process be like?